International Journal of Mechanical and Production
Engineering Research and Development (IJMPERD)
ISSN (P): 2249-6890; ISSN (E): 2249-8001
Vol. 7, Issue 5, Oct 2017, 111-116

DMATM: DUAL MODIFIED ADAPTIVE TECHNIQUE BASED MULTIPLIER

R GANGADHAR REDDY[1], UZMA FATHIMA[2], P. B. NATARAJAN[3] & M. RAMESH BABU[4]

[1,2,4]*Department of Electronics and Communication Engineering, Institute of Aeronautical Engineering, Hyderabad, Telangana, India*

[3]*Department of Electronics and Communication Engineering, MLR Institute of Technology, Hyderabad, Telangana, India*

ABSTRACT

The most critical arithmetic functional units are digital multipliers. This system overall performance is depends on the throughput, of the multiplier. An n-bit by n-bit multiplier can be employed by an n-bit by n-bit multiplication, which can be carried out iteratively. K cycles are required for completion of a multiplication, which can be terminated earlier, if some of the leading significant bytes are all zeros or ones. This paper proposes a simple scheme of dual modified adaptive, to exchange the two operands dynamically, for reducing more cycles to 32-bit by 32-bit multiplications. The designing of a dual modified adaptive multiplier for 32*32 bit number multiplication is proposed. Now-a-days, computer systems are very high speed unique multiplier. Hence, a Dual Modified adaptive multiplier is proposed. M, N, interconnected blocks are generated by this proposed system. The Dual Modified adaptive multiplier is obtained, by enlarging bit of operands and producing an additional product. Multiplication operation which is performed by Dual Modified adaptive is efficient with less area and it reduces the delay.

KEYWORDS: K Cycles are Required 32-bit by 32-bit Multiplications & Reduces the Delay

Received: Aug 04, 2017; **Accepted:** Aug 23, 2017; **Published:** Sep 02, 2017; **Paper Id.:** IJMPERDOCT201712

Original Article

INTRODUCTION

The important task for many microprocessors and **DSP** processors is to design a cost effective multiplier. For achieving the major objectives are area, delay and power consumption. Parallel implementations can use more area and consume more power but which result in a fast multiplier. The serial implementations are using less area and consuming less power, but result in a slow multiplier. A 32-bit by 8-bit Boothmultiplieris employed and spends **4** cycles, for carrying a32-bit by 32-bit multiplication. This implementation signifies a balance among power, delay and area. It requires more cycles to perform a multiplication, but it takes much less time for each cycle and requires less area than a 32-bit by 32-bit parallel multiplier.

In order to perform correctly, traditional circuits use critical path delay as the overall circuit clock cycle. However, the probability that the critical paths are activated is low. In many cases, the path delay is shorter than the critical path.

The critical path delay is used as the overall cycle period, which will result in significant timing waste. Hence, to reduce the timing waste of traditional circuits, the variable-latency design was introduced. Booth multiplier and thereby minimizes the effect of glitches. The extra delay and power consumption can be minimized, if the exchange logic is made simple. Note that, from the operand source registers to the multiplicand and ***multiplier,*** registers the extra delay is added to that path.

If this path is not originally a critical path, then extra delay would have a limited influence on the

multiplier performance. Although, if the path is a critical one, then exchanging of the two operands can be performed subsequently, when the multiplicand and multiplier registers are loaded with the two operands. But this change in the architecture would make the glitches caused by the exchange logic, propagate through the multiplier. All these techniques perform the exact computation and modules, to produce the correct result. Accuracy of the module devices is always 100%, in exact computing. But exact computing has one major drawback. It is impossible to optimize all the parameters of the circuit in exact computing. However, exact computing is not essential for every application. There are some applications like image processing and multimedia, which can tolerate errors and provide meaningful results.

Inexact (approximate) computing techniques have become popular, because of its low complexity and less power consumption. Inexact computing produces reasonable result, even if it has low accuracy. In approximate computing, the value of error rate(ER), error distance (ED) and normalized error distance (NED), play an important role to calculate the final output. Error rate is given by a number of erroneous outputs, over the total number of outputs. Error Distance is the arithmetic distance among an erroneous output and the correct one. Normalized Error Distance is the ratio of mean error distance, over all inputs by maximum input of the circuit's approximation techniques, for adders and multipliers.

The normal addition rule is applied in accurate part, whereas a special method of addition takes place in inaccurate part. Output "sum" value is calculated normally, when any one of the operand value of adder is "0". When both operands are"1", "sum" value can be fixed as "1" from that bit position, to least significant bit. This technique is used to minimize the error distance of the adder, in load demand and renewable generation, because of day-ahead forecast error, there exists inevitable deviation between real time operation and day-ahead generation schedules. Especially in the recent years, the increasing penetration of intermittent resources aggravated the situation and confronted the system with high variability and uncertainty. This poses great challenges on the secure and reliable operation of the power systems. One alternative to solve this problem is, to introduce intra-carry generation scheduling (ICGS), which fills the gap between day-ahead generation schedules and real-time operation, and serves as an effective tool for optimal dispatch over a look-ahead time horizon. By enforcing ICGS, we can re-schedule the conventional generation units, with the latest forecast information to meet the net load in a more cost-effective way.

LITERATURE SURVEY

When shorter paths are frequently activated, then the average latency in the variable-latency design is better than that of traditional designs. For example, speculation techniques are used by many variable-latency adders with error detection and recovery. In addition, the critical paths are divided into two shorter paths that could be unequal. The clock cycle is set to the delay of the longer one. To improve performance, these research designs were able to reduce the timing waste of traditional circuits, but they did not consider the aging effect. During the runtime, it could not adjust themselves.

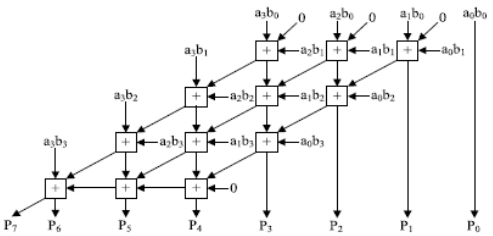

Figure 1: 4x4 Normal Array Multiplier

An improvement method of the normal array multiplier (AM), is a Column By-passing Multiplier. The AM is a fast parallel multiplier and is shown in above Figure 1. The multiplier array contains (n−1) rows of carry save adder (CSA). Each row contains(n − 1) full adder (FA) cells. In the CSA array, each FA has two outputs: 1) the sum bit goes down and 2) the carry bit goes to the lower left FA. The last row is a ripple adder for carry propagation. In the AM the FAs are always active without regarding of input states. In a low-power column-by passing multiplier design, the FA operations are disabled if the corresponding bit in the multiplicand is 0.

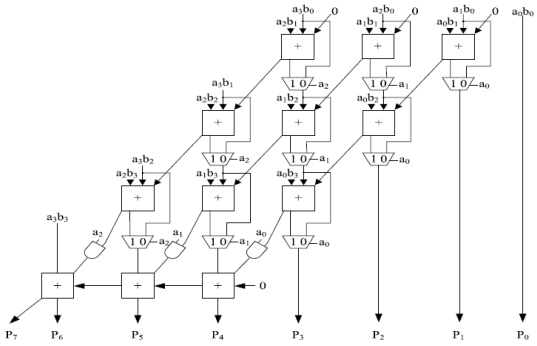

Figure 2: Column-Bypassing Multiplier

The above Figure 2 shows a 4×4 column-bypassing multiplier. For example, let us assume the inputs are10102 * 11112, it can be noted that for the FAs in the first and third diagonals, two of the three input bits are 0. Therefore, in both diagonals the output of the adders is 0, and the output sum bit is simply equal to the third bit, which is the sum output of its upper FA.

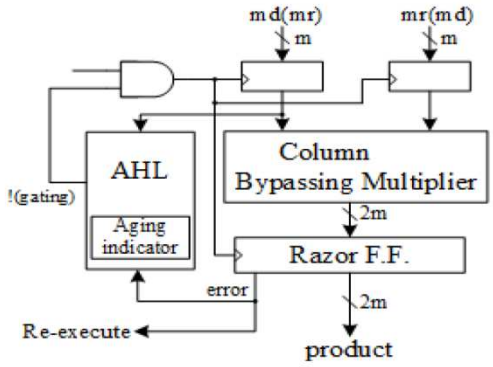

Figure 3: Existed System Block Diagram

In the existed system architecture, the column- and row-bypassing multipliers can be examined by the number of zeros in either the multiplicand or multiplicator. It is to predict, whether the operation requires one cycle or two cycles to complete. When there are random input patterns, the number of zeros and ones in the multiplicator and multiplicand follows a normal distribution.

Hence, by using similar architecture the two aging-aware multipliers can be implemented. The difference between the two bypassing multipliers, lies in the input signals of the AHL. The input signal of the AHL in the architecture, with the column-bypassing multiplier is the multiplicand, whereas that of the row-bypassing multiplier is the multiplicator. Razor flip-flops can be used, to detect whether timing violations occur before the next input pattern arrives. Figure 3 shows the details of Razor flip-flops. A main flip-flop, shadow latch, XOR gate and mux are included in a 1-bit Razor flip-flop.

The main flip-flop catches the execution result, by using a normal clock signal. The shadow latch catches the execution result, using a delayed clock signal, which is slower than the normal clock signal. If the latched bit of the shadow latch is different from that of the main flip-flop, then the path delay of the current operation exceeds the cycle period and the main flip-flop catches a result which is incorrect. When errors occur, the Razor flip-flop will set the error signal to 1, to notifying the system tore execute the operation and notify the AHL circuit that an error has occurred. Razor flip-flops are used to detect whether an operation, which is considered to be a one-cycle pattern can really finish in a cycle. If not, the operation is re executed with two cycles. Although the re execution may seem costly, if the re execution frequency is low, then overall cost is low. In the aging-ware variable-latency multiplier, the key component is AHL circuit. The AHL circuit consists of an aging indicator, two judging blocks, one mux, and one D flip-flop. The aging indicator used for indicating whether the circuit has suffered significant performance degradation, is due to the aging effect. In a simple counter, the aging indicator is implemented to counts the number of errors over a certain amount of operations and it is reset to zero at the end of those operations. The column- or row-bypassing multiplier is not able to complete these operations successfully, if the cycle period is too short, causing timing violations.

PROPOSED SYSTEM

According to the input logics, the gates of in the Dual Modified adaptive multiplier are always active. The operations in the Dual Modified adaptive multiplier design are disabled, if the corresponding bit in the multiplicand is 0.

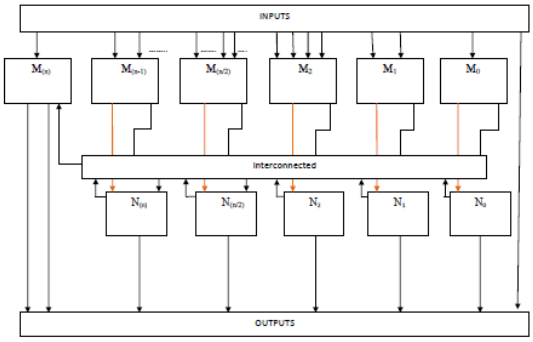

Figure 4: 4x4 High Performance Dual Modified Adaptive Multiplier

In above figure $M_0, M_1, M_2, \_ \_ \_, M_n$ have done their operation, and outputs are given to the interconnected block and N- block, simultaneously. Depending on the priority in the operation, the Dual modified adaptive gives the N-block output to interconnected block and vice versa. Hence, in both diagonals, the output of the adders is 0 and the sum bit which is output equal to the third bit. Figure 4, shows the 4*4 high performance Dual Modified adaptive multiplier, which uses the Critical path cycle as an execution cycle period. Dual modified adaptive multiplier reduces the timing waste, which occurred in traditional circuits. We can execute a shortest path by using Dual modified adaptive. The architecture is extended up to 32*32 bits.

Dual Modified adaptive is extensively acquired in multipliers, because it can reduce the number of partial product rows to be added. Thus, Dual Modified adaptive is used for reducing the size and enhancing the speed of the reduction tree. The least significant bit position, of each partial product row encoding gives an irregular partial product array and a complex reduction tree. Hence, the Dual Modified adaptive multipliers, with partial product array produce a very high speed.

RESULTS

Figure 5: RTL Schematic of Dual Modified Adaptive Multiplier

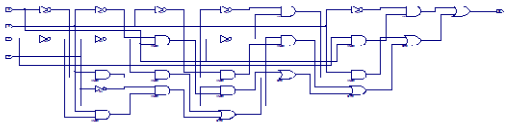

Figure 6: LUT in Technical Schematic of Dual Modified Adaptive Multiplier

Figure 5, shows the RTL schematic and Figure 6, shows the Look Up Table (LUT) in technical schematic of high performance, of the Dual Modified adaptive multiplier.

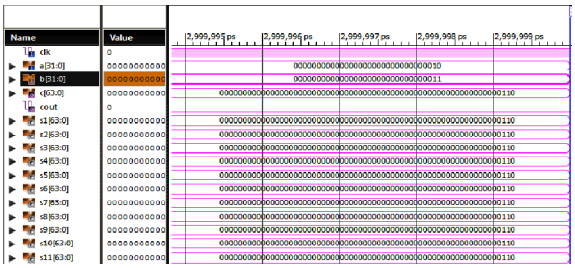

Figure 7: Output Waveform

Table 1: Comparision Table

	Memory Occupied (KB)	Delay (ns)
EXISTED SYSTEM	497	134
PROPOSED SYSTEM	380	104

CONCLUSIONS

An aging-aware variable-latency multiplier is designed with the AHL multiplier. It is able to adjust the AHL to mitigate performance degradation, due to increased delay. The experimental results show the proposed architecture with 32*32 column-bypassing multipliers. This paper proposes a simple scheme, to exchange the multiplicand and multiplier dynamically to reduce more execution cycles for 32-bit by 32-bit multiplications, with a small area and delay overhead. The Dual Modified adaptive can generate the M and N, interconnected blocks. Each block contains gates and row in the architecture. Dual Modified adaptive multiplier is more efficient than the existed system. The required hardware and chip memory is reduced, by the proposed system. It reduces the delay.

REFERENCES

1. *D. Liu Embedded DSP Processor Design, 1st ed. Morgan Kaufmann Publishing, 2008.*
2. *K. K Parhi VLSI Digital Signal Processing Systems: Design and Implementation.,1st ed. John Wiley and Sons, 1999.*

3. *Y. Kim, Y. Zhang, and P. Li, "An energy efficient approximate adder with carry skip for error resilient neuromorphic VLSI systems," in proc. of International conference on Computer-Aided Design (ICCAD), Nov. 2013, pp. 130-137.*
4. *A. Pishvaie, G. Jaberipur, and A. Jahanian, "Improved CMOS (4; 2)compressor designs for parallel multipliers," Computers and Electrical Engineering, vol. 38, no. 6, pp. 17031716, Nov. 2012.*
5. *D. Baran, M. Aktan, and V. G. Oklobdzija V. G, "Energy Efficient Implementation of Parallel CMOS Multipliers with Improved Compressors, "in proc. ACM/IEEE International Symposium on Low-Power Electronics and Design (ISLPED), Aug. 2010, pp. 147-152.*
6. *S. Veeramachaneni, K. Krishna M, L. Avinash, S. R. Puppala, and M. B. Srinivas, "Novel Architectures for High-Speed and Low-Power 3-2, 4-2 and 5-2 Compressors," in proc. of International Conference on VLSI Design (VLSID), Jan. 2007, pp. 324-329.*
7. *R. Menon, and D. Radhakrishnan, "High performance 5: 2 compressor architectures," in proc. of IEE - Circuits, Devices and Systems, vol. 153, no. 5, Oct. 2006, pp. 447-452.*
8. *A. Calimera, E. Macii, and M. Poncino, "Design techniqures for NBTI tolerant power-gating architecture," IEEE Trans. Circuits Syst., Exp. Briefs, vol. 59, no. 4, pp. 249–253, Apr. 2012.*
9. *K.-C. Wu and D. Marculescu, "Joint logic estructuring and pin reordering against NBTI-induced performance degradation," in Proc. DATE,2009, pp. 75–80.*
10. *Y. Lee and T. Kim, "A fine-grained technique of NBTI-aware voltage scaling and body biasing for standard cell based designs," in Proc. ASPDAC,2011, pp. 603–608.*
11. *M. Basoglu, M. Orshansky, and M. Erez, "NBTI-aware DVFS: A new approach to saving energy and increasing processor lifetime," in Proc. ACM/IEEE ISLPED, Aug. 2010, pp. 253–258.*
12. *K.-C. Wu and D. Marculescu, "Aging-aware timing analysis and optimization considering path sensitization," in Proc. DATE, 2011, pp. 1–6.*
13. *K. Du, P. Varman, and K. Mohan ram, "High performance reliable variable latency carry select addition," in Proc. DATE, 2012, pp. 1257–1262.*
14. *A. K. Verma, P. Brisk, and P. Ienne, "Variable latency speculative addition: A new paradigm for arithmetic circuit design," in Proc. DATE,2008, pp. 1250–1255.*
15. *D. Baneres, J. Cortadella, and M. Kishinev sky, "Variable-latency design by function speculation," in Proc. DATE, 2009, pp. 1704–1709.*
16. *Y.-S. Su, D.- C. Wang, S.-C. Chang, and M. Marek-Sadowska, "Performance" optimization using variable-latency design style," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 19, no. 10, pp. 1874–1883, Oct. 2011.*